

Lotus Development Corporation's 1-2-3

A fast, versatile package that combines spreadsheet, graphing, and database functions

Gregg Williams
Senior Editor

When does "evolution" become "revolution"? When I first saw 1-2-3, a spreadsheet/database/graphing system from Lotus Development Corporation, I thought, "Hmm, very well done, but it's just an extension of existing software." True enough. But after using the product for a few hours, I realized it impressed me as more than just an evolutionary product. There may be nothing new under the sun, as they say, but there *are* novel ways to combine old things. In that sense, Lotus's 1-2-3 is modestly revolutionary because it synergetically combines three packages. In this product description, we'll take a look at the system's capabilities.

Spreadsheet Capabilities

1-2-3 is, above all else, a spreadsheet. Like most spreadsheets, it lets you enter either text, numbers, or formulas in a network of "cells" so that, by changing the content of certain cells, you can perform an involved set of calculations automatically. It's safe to say that 1-2-3 has all the features you've ever seen on spreadsheets. You can copy ranges of cells, insert and delete rows and columns, change the output format of a range of cells or the width of a column of cells, and do numerous other functions.

The size of the spreadsheet is 2048 rows of 256 columns. Lotus claims that 1-2-3 will handle up to 640K bytes of memory. You can't fill the entire spreadsheet with that, but it's probably considerably more than enough for most applications.

1-2-3 will soon be available for the IBM Personal Computer (PC) and will eventually be converted to other

microcomputers that use the Intel 8086 or 8088 microprocessor. The initial version of 1-2-3 will need an IBM PC with two disk drives, 128K bytes of memory, and either a monochrome or a color display; if the computer has both, you can view the spreadsheet (on the monochrome monitor) and graphs (on the color monitor) at the same time. If you have only the monochrome video display, you cannot view your graphs; you can only print them out. If you have only the color video display, you can alternate between viewing the spreadsheet and the graph.

Graphing Capabilities

1-2-3's sophisticated graphing commands enable you to create graphs of up to four variables using information already on the spreadsheet. Photo 1a shows a small spreadsheet; photos 1b and 1c show the two graphs of the same data. You can ask for one of five kinds of graphs, including bar and line graphs (of which photo 1b is an example), a pie chart (of one variable only), a stacked bar chart, or an x-y graph (two lists of variables used as x-y coordinate pairs). During my first session, I set up the parameters for a graph in under three minutes; after a few tries, I could do it in less than a minute. The graph is drawn in under two seconds—a far cry from graphing, say, on the Apple II.

Once you've made a graph, three keystrokes will display it in another form; if data in the spreadsheet has been changed, you can display a revised graph with one keystroke. Various options let you change the look of a graph; you can display one in black-and-white if you don't have a color monitor attached to the color video in-

1a

1-2-3 (F8) 209389

ENTER

Circulation of Magazines

| | BYTE | Gerbil Illustrated | Popular Bag Lady |
|------|--------|--------------------|------------------|
| 1/82 | 245338 | 18381 | 3829 |
| 2/82 | 244325 | 21987 | 8392 |
| 3/82 | 251861 | 49278 | 17392 |
| 4/82 | 272287 | 283837 | 59273 |
| 5/82 | 279571 | 58766 | 183182 |
| 6/82 | 289888 | 28198 | 193872 |

1b

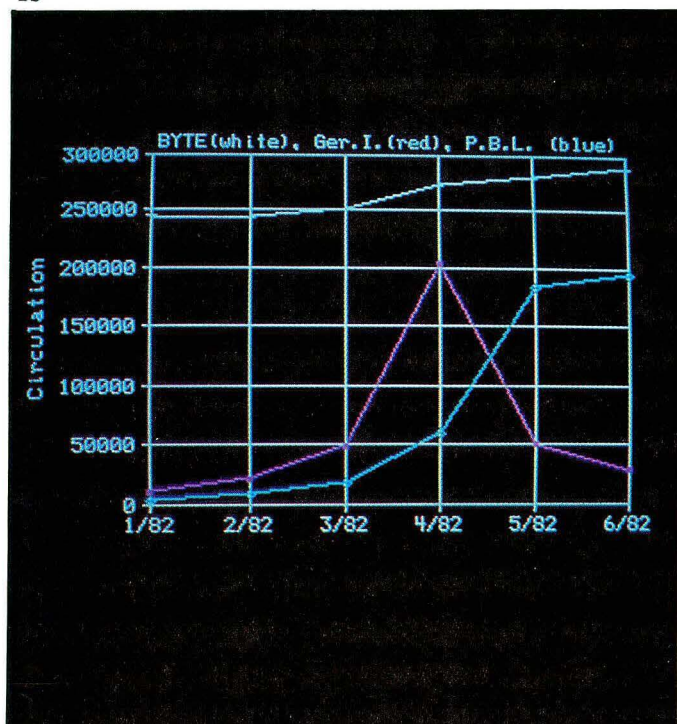


Photo 1: Making graphs from spreadsheet data. Given the small spreadsheet shown in photo 1a, the line graph in photo 1b and the bar graph in photo 1c were both made from the data on the spreadsheet. Once the data to be used has been specified, a different type of graph can be drawn (or a graph with new data can be redrawn) in less than two seconds.

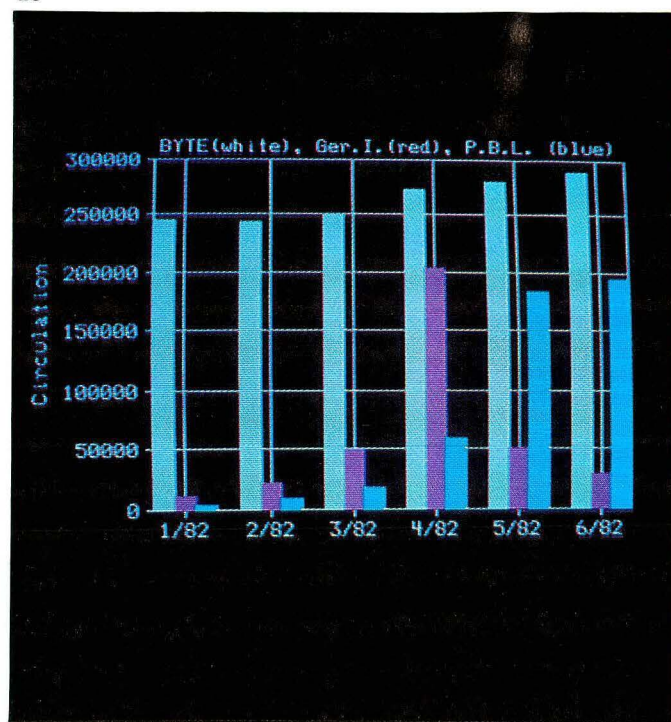
terface. You can also send a graph to the printer; at the moment, only the Epson MX-80 printer is supported, but others will be supported in the final version.

Database Capabilities

You can also use 1-2-3 as a database for storing, sorting, and retrieving records. Although its database capabilities are not comparable to those of, say, dBASE II, they are very useful in conjunction with the other two.

1-2-3 will take an arbitrary area of the spreadsheet to be a database; the entries on a given row are considered to be a *record*, and record fields must be vertically

1c

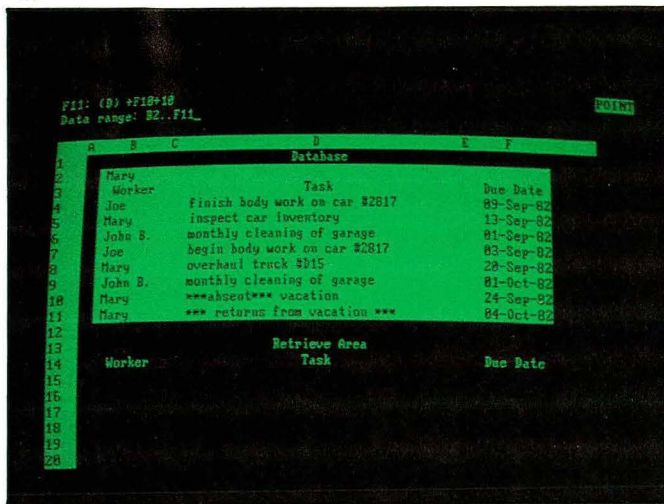


aligned. (The database can be anywhere on the spreadsheet along with nondatabase information.) You can sort a set of records, query it, or use it to retrieve selected records. Records are sorted by a maximum of two keys, each of which specifies a sort operation by either ascending or descending order. The query and retrieve operations are very similar. Both find records that match certain criteria; the former modifies the action of the cursor-up and cursor-down keys so that the cursor will highlight only records that match the criteria, and the latter copies the matching records into a designated area of the spreadsheet.

Photo 2a shows a small collection of records that is being readied for a retrieve operation. 1-2-3 prompts you for the area of the spreadsheet that is considered a collection of records; when you choose that area, 1-2-3 highlights it in reverse video. The top line of the database area contains the values (or, in the case of inequalities, the relationships) you're searching for, the second line is the name for each field, and the lines below that are the actual records. Once the retrieve function is completed, the matching records are deposited in the assigned destination area (see photo 2b).

Granted, 1-2-3's database capabilities don't match those of the expensive databases, so it would be foolish to buy the system in lieu of a full-featured one. But *selection* is a fundamental data-manipulation operation, and any package that can speak to this need is superior to those that don't. I suspect that the database functions in 1-2-3 will be used most often to isolate specific data that will then be graphed (if you had to isolate the data manually, you probably wouldn't bother graphing it). Still, 1-2-3's database can be used in several traditional applications as well as in some less traditional ones—scheduling, for example.

2a



2b

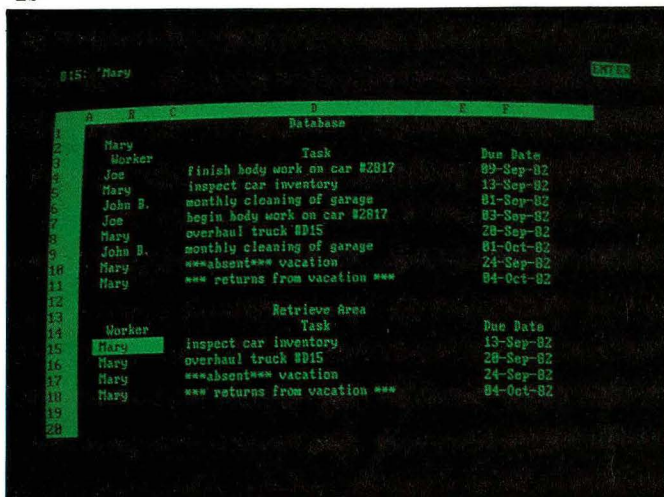


Photo 2: The retrieve function in 1-2-3. If we consider a series of spreadsheet rows to be records of a file, we can select certain records based on given criteria. In photo 2a, we are setting up a database of records (shown in inverse video) to be queried for all the records that have a name field of "Mary"; 1-2-3 will place the matching records in a separate area in the bottom half of the screen, as shown in photo 2b.

The Computing Power of 1-2-3

We've seen enough spreadsheet programs to expect each new one to improve on the previous versions. In this case, the program fulfills our expectations; 1-2-3 is the product of a few very creative minds. The program was designed by Mitch Kapor and Jonathan Sachs and programmed by Sachs, George Riner, and Rick Ross. Sachs worked on four spreadsheet programs before this, and 1-2-3's easy-to-use design reflects his experience.

Numbers alone don't tell the full story, but they'll have to do here. Visicorp's Visicalc for the IBM Personal Computer has 15 arithmetic, logical, and relational operators, 28 functions, and 32 commands. 1-2-3 has 15 operators, 41 functions, and 66 spreadsheet-related commands. If we include the database and graphing commands, 1-2-3 actually has 110 commands.

Although several features of the spreadsheet module are new to this product, two stand out. First, 1-2-3 allows you to manipulate both spreadsheets and their printed versions (called print files). That means you can save or combine parts of either the spreadsheet itself (including the formulas in each cell) or its printed representation (the letters and numbers in each cell).

The second set of especially noteworthy commands, /DT1 and /DT2, is for creating tables. With them, you can automate the tedious work of charting the behavior of a spreadsheet when one or two input variables vary across a given range of values. You can specify either one or two lists of input arguments, the cell positions in which these values are to be entered, and the cell that will contain the desired output. 1-2-3 will then substitute the input values into the spreadsheet and accumulate the output values in a one- or two-dimensional table. Photo 2a shows the format of the two input ranges (one a column and the other a row) and the corner of the spreadsheet that performs the calculation (for illustrative purposes, the ratio of the two input values). Photo 2b shows the table created by the execution of the /DT2 two-dimensional table-creation command.

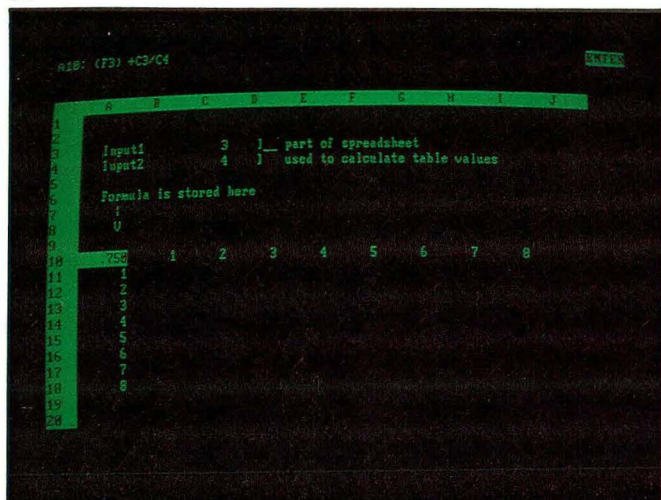
Human Engineering

1-2-3 also excels in human engineering factors, those elements of a program that make it easy to use. I cannot overemphasize the importance of human engineering in microcomputer programs. To date, computers have been hard to understand and inconvenient to use, which has discouraged many people from using them. 1-2-3 is one of the few pieces of software that can literally be used by anybody. You can buy 1-2-3 and an IBM Personal Computer and be using the two together the same day.

1-2-3 is one of a new breed of advanced software products that has a disk-based help file. At any time, you can hit a Help button (the F1 key on the IBM PC) and get one or more screens of information on literally any aspect of the program. The help file, which resides on the A disk of the IBM PC, contains over 250 screens of information. When you hit the F1 key, a menu of topics appears on the screen in under one second (an important consideration if waiting to see disk-based information discourages you from using the Help key). You then use the IBM PC arrow keys to move an inverse-video cursor to the desired topic and press the Return key. Each screen is cross-indexed to related screens and to a main menu, and you can get to any screen in less than 15 seconds. I was able to find the information as quickly as if I had looked it up in the 1-2-3 documentation. Those who are new to computers will be very impressed with this feature and will be much more inclined to use 1-2-3 because of it. (Software Arts' TK Solver also uses a disk-based help file, and I understand future products from Visicorp will include them as well.)

1-2-3 is, to my knowledge, the first spreadsheet program to distribute a comprehensive tutorial package that teaches the beginner how to use it. Software producers

3a



3b

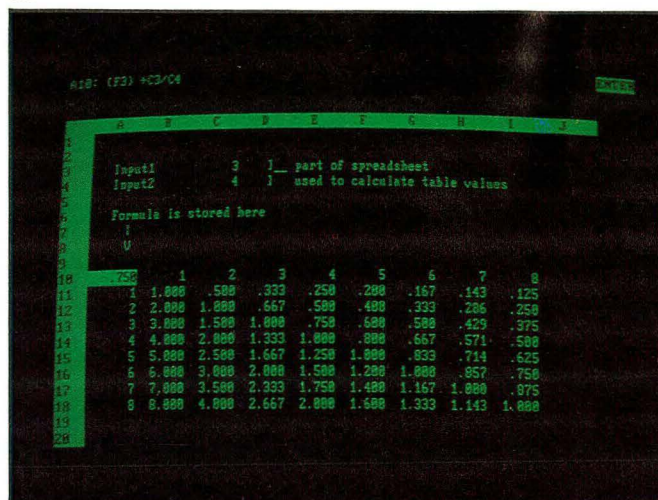


Photo 3: Two-dimensional table creation in 1-2-3. When it's given a spreadsheet, two lists of input values, input cells for both lists, and an output cell, 1-2-3 can automatically create a table that gives the output value for each combination of the two input values. Photos 3a and 3b show the spreadsheet before and after the command is given. The output function shown here is the ratio of the two input values.

have always deliberated over the best way to teach a buyer (regardless of his understanding of the computer) how to use a complicated product. Lotus has incorporated these interactive tutorial programs into 1-2-3. They are reputed to cover, in some depth, the entire program. The segment I saw does its job well; it interactively shows you around the IBM PC keyboard and tells you how to move the cursor around a spreadsheet. This set of programs is, in effect, an indefatigable instructor who is always ready and willing to show you how to use the product and who will never laugh at your mistakes. This is another very strong incentive for the beginner to buy 1-2-3. No matter what your level of expertise, it's a very nice feature.

Another feature that illustrates 1-2-3's human-engineering design is its jargon-free prompts, explanatory messages, and fail-safe mechanisms. Suppose I want to delete a spreadsheet file. When I type "/F", I get a menu of possible file actions, the first of which is in inverse video. I can execute any action by either moving the inverse video cursor to that action name and hitting the Return key or hitting the key that is the first letter of the action name. If I hit the right-arrow key until the action name "Delete" is in inverse video, I get an explanatory note immediately below the command line that says, "Delete a worksheet, print, or graph file" ("worksheet" is Lotus's name for a spreadsheet). That's what I want to do, so I conveniently hit Return. The inverse video cursor is already positioned on the option I want, "Worksheet", so I hit Return again. 1-2-3 now reads the current disk and gives me a menu of all the spreadsheet files by name. I then move the inverse video cursor to the spreadsheet I want to delete and hit Return again (there's no need to type the file name—just point to it). 1-2-3 displays two options, "No" and "Yes", with the cursor on the "No" option and an explanatory note, "Do not delete the file". By

moving the cursor onto the word "Yes" (which includes the note "Delete the file") and pressing Return, I can delete the spreadsheet file. Wouldn't you like a piece of software that does all that for you?

1-2-3 gives you a lot of visual feedback on your choices. The inverse-video cursor and menu system described above is one such example. (The resemblance to Visicorp's Visiplot package is not accidental; Mitch Kapur, who designed Visiplot, is the president of Lotus.) Another example is 1-2-3's ability to let you specify coordinates by moving the cursor to them instead of listing them by row-column designation. To specify an area of the spreadsheet, you simply move the cursor to a corner of the area with arrow keys, "tack" it in place (usually with the same "." command used in Visicalc), and move the cursor away from that point. A rectangular area that spreads from the current cursor location to the "tacked" location appears in inverse video (see photo 2a).

Pressing a previously defined macro key causes the equivalent string to be executed as if it had been typed in from the keyboard—a feature with great potential.

Pointing to both spreadsheet locations and menu options makes 1-2-3 very easy to use and reduces errors.

Named ranges are another way of pointing to an area of memory. Any cell or rectangular area of the spreadsheet can be given a name of up to 15 characters. That name can then be used wherever the cell or range coordinates are needed.

dinates would usually be used. Sometimes, a named range is simply a convenience; in other places (e.g., in the formula for the value of a given cell), it makes the spreadsheet more readable. Microsoft Consumer Products' Multiplan automatically assumes that you can refer to a cell by the text label immediately to its left; 1-2-3 stipulates that you create a named range, but that range is more versatile than a named cell in Microplan. (People at Lotus have told me that the final version of 1-2-3 has an option that lets you refer to a cell automatically by an adjacent label value; you have the choice of positioning the labels below, above, or to the left or right of the cells.)

1-2-3 uses the ten IBM PC function keys (on the left side of the keyboard) in two ways. The unshifted keys are used for ten often used 1-2-3 commands; for example, F1 is the Help key described above, F5 is a Goto key that moves the spreadsheet and cursor to a given location, and F10 is a Redraw command for the most recently drawn chart. The ALT key pressed simultaneously with a letter key gives you 26 user-defined macro keys. You can define each of the 10 available macro keys to be any string of characters you desire, including the unshifted function keys, the arrow keys, and the Return key. When pressed, the macro key causes 1-2-3 to execute the equivalent string as if it had been typed in from the keyboard. This feature has great potential; in certain situations, you may be very glad not to have to retype the same keystrokes repeatedly.

1-2-3 has what I call "intelligent" labels, text strings that are not influenced by the current cell width. Say, for example, I want to print a 30-character title across a spreadsheet, and assume that all the cells are 8 characters wide. In many spreadsheet programs, you'd have to split the title manually and enter it in four 8-character chunks—awkward and tedious, right? In 1-2-3, though,

1-2-3's ability to "protect" cells means that data cannot be destroyed unintentionally.

you enter the full title in the first cell you want it to occupy. The string itself stays in that cell as its formula, but in its display it spills over into the space normally used by cells to the right. The effect is the same as with other spreadsheets, but it is achieved much more easily. What if you decide your cells must be 10 characters wide? In other spreadsheet programs, the title is mangled when you change the cell width—each 8-character chunk has two blank characters to its right. But because the value of the string is retained in only one cell, 1-2-3 displays the title correctly despite the change in cell width. Several text-justification commands in 1-2-3 also help format a single string into rectangular spreadsheet areas that occupy one or more rows of cells.

One final human-engineering feature of 1-2-3 is its ability to *protect* cells—that is, to keep you from assigning new values to cells or ranges of cells. When a spread-

sheet is used by anyone but its designer, protected cells allow the person to use it without inadvertently destroying valuable data or formulas. If you try to write a new value into a protected cell, you will get an error message that says the cell is protected. This is a nice feature, but it would be more useful if cells could be "locked" (so you couldn't cancel the protection of a cell) and made invisible, features that are available in Visicorp's Visicalc Advanced Version for the Apple III computer.

Speed and Integration Advantages

In addition to being powerful and easy to use, the various modules of 1-2-3 are fast and well integrated.

Many software developers (Software Arts, Visicorp, and Microsoft Consumer Products, to name a few) are writing their software in high-level languages that are usually compiled to the native code of the machine's microprocessor. They do that in order to move a given program to more than one machine and thus maximize their profits. (Rightly so—good software is very expensive to create.) An interesting side effect of using high-level languages is that the resulting product is slower than if it had been written strictly for the native microprocessor. Depending on the efficiency of the high-level language used, an assembly-language version of a product can be considerably faster and more compact than its high-level counterpart. This is certainly the case with 1-2-3; it is coded in highly optimized 8086 assembly-language code. Granted, I have only Lotus's word that it is "highly optimized," but 1-2-3 is fast—I didn't have to wait when I expected to. [Editor's Note: A demonstration of 1-2-3's capabilities at its recent public unveiling illustrated the program's speed. For example: a spreadsheet was displayed showing a listing of 25 hotels ordered by location. To the right of the hotel listings were 12 or so columns of data showing vital statistics for the hotels. After setting up the appropriate initial conditions, the program was able, within five seconds, to re-sort the hotels by revenue, calculate averages for key pieces of data, reinsert them into the model, add compensating factors for possible future inflation, calculate projected revenue figures for the next few years, and graph the results. . . . C. M.]

Integration is a very important characteristic of 1-2-3. Because the spreadsheet, database, and graphing programs are in the computer simultaneously (1-2-3 does not use overlays to bring in sections of code when called), you are more likely to use them. I for one am always annoyed when I have to wait for UCSD Pascal to load another part of the language system whenever I go, say, from the Filer to the Editor. I would be less apt to experiment with graphing different sets of data with Visicorp's Visicalc and Visiplot, which would involve saving my data to disk, exchanging disks, starting up the Visiplot program, exchanging disks again, reading in the data, and, finally, plotting the data. I would do a similar sequence of disk and program switching to get back to



Visicalc and adjust my data. How much experimentation does that rigamarole encourage?

Caveats

This product description is based on more than 20 hours of experimentation with a version of 1-2-3 Lotus loaned to me a month before the design and code were finalized. I experimented with all of the features mentioned in this article and several others. I found 1-2-3 easy to use; it didn't mislead me into doing something I didn't want to do or leave me confused as to my place within the structure of the program. Although I didn't have a copy of the final documentation, I found that a functional specification document I was given and the disk-based help file provided all the information I needed. I did have trouble specifying a new range by pointing in some situations; Lotus told me that this error will be corrected before the product is available commercially.

News from Lotus

1-2-3 will be available for the IBM Personal Computer sometime next month; it will eventually be available for other 8086- and 8088-based microcomputers, although Lotus has announced no definite plans or machines. Lotus has also fixed the price of 1-2-3 at \$495, which makes it a tremendous buy for the money. Staff members point out that 1-2-3 improves on the Visicorp trilogy of Visicalc, Visiplot, and Visidex (which together sell for a total of \$700 in their IBM PC versions) in both price and capabilities.

Mitch Kapor and his team of designers and programmers are incredibly creative: they have come up with more good ideas than they can possibly implement at one time, but they also implement more of them than I would

have thought possible. They have indicated that 1-2-3 will probably be able to read dBASE II files, thus making it possible for 1-2-3 to interchange data with one of the most popular databases around. They also told me that 1-2-3 will be able to make the spreadsheet look like a business form with blanks to be filled in by the user, thus enabling you to enter data into 1-2-3 database areas.

Lotus plans to add several graphics features to the final version. These include visual superimposition of charts, the use of text in user-chosen shapes, sizes, and colors; choice of printed chart size; manual scaling of graph axes; and support of the Hewlett-Packard HP7470A plotter and several popular printers. In addition, Lotus plans to add word-processing capabilities to subsequent releases of 1-2-3.

Conclusions

On the basis of the prerelease version of the software, 1-2-3 promises to be a fast, easy-to-use, integrated package for people who need to manipulate numbers, graphs, and records of data. Its instantly available help file, interactive tutorial programs, and incorporation of tested human-engineering concepts make it particularly impressive. It is one of the first of a new breed of sophisticated applications software that is both powerful *and* easy to use, even for beginners.

I'm very pleased about 1-2-3's price of \$495. That puts it within the grasp of both the professional who needs a sophisticated spreadsheet program and the individual who wants one for personal use.

In any case, even unfinished, 1-2-3 is a fine piece of software. I look forward to seeing the first and subsequent versions of it. ■

